REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE March 9, 2005	3. REPORT TYPE AND DATES COVERED Final Technical Report July 10, 2003 through March 9, 2005

4. TITLE AND SUBTITLE
Unique Signature Detection Program

5. FUNDING NUMBERS
DAAD19-03-C-0070

6. AUTHOR(S)
James H. Raymer, Jun Liu, Ye Hu, Larry Michael, Shiying Wu, Joey Morris, and Michelle McCombs.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
RTI International
Attn.: Dr. James H. Raymer\
3040 Cornwallis Rd
Research Triangle Park, NC 27709

8. PERFORMING ORGANIZATION REPORT NUMBER 08873 FR

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
45509.1-LS-OD

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
An approach to quantitatively describe the relationship between the volatile organic chemical profile associated with human emanations, as measured by gas chromatography/mass spectrometry, and genetic composition, specifically the HLA complex, was developed. To reduce random noise in the analysis, variable selection was carried out. Subsequently, two statistical analysis methods were evaluated and used to further eliminate those elements that did not significantly contribute to the distinction of the genotypes. These methods were: analysis of variance and Stepwise Linear Discriminant Analysis (SLDA). Not surprisingly, the latter was found to be superior because it takes into account the data covariance structure. For classification of the chemical profiles, several linear and nonlinear discriminant analyses were evaluated. These results showed that Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM) are preferred among the linear and nonlinear classification methods, respectively. RTI methods successfully classified individual samples into the correct genotype 90% of the time when the number of genotypes was relatively small (less than 10). The recommended approach is SLDA to select important components followed by LDA when the sample size is small and SVM when the sample size is moderate or large for classification purposes.

14. SUBJECT TERMS
Data processing, data analysis, data coordinating center, Gas chromatography/mass spectrometry data, statistical analysis, classification

15. NUMBER OF PAGES
14

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

Unique Signature Detection Program

Final Report Statistical Analysis Approach

March 9, 2005

Prepared for:

DARPA and ARO
P.O. Box 12211
Research Triangle Park, NC 27709-2211

Prepared by:

RTI International
3040 Cornwallis Road
Research Triangle Park, NC 27709

RTI Project Number: 8873

Table of Contents

Page

Table of Contents (Continued)

List of Tables

1.0 Introduction

The overall objective of this Unique Signature Detection (USD) Program is to relate the volatile chemical signature of human emanations to genetic composition of the MHC complex to determine if the chemical signature can uniquely identify individuals. RTI's role in the larger program is to conduct a limited field study to investigate the relationship of MHC to volatile organic chemicals (VOCs) in sweat and to perform an independent statistical analysis of data generated by the three other research teams (Monell/Battelle, Draper Laboratory, Konrad Lorenz Institute). The specific goals of the project at RTI are:

1. To identify and measure the concentrations of volatile organic chemicals (VOCs, unique chemical signature) in sweat,

2. To investigate whether the VOCs are specific to an individual,

3. To investigate how individual chemical signature is expressed by an examination of the relative concentrations of the volatiles or the presence/absence of the volatiles,

4. To investigate the relationship between odor type with MHC (Human HLA), and

5. To devise a Statistical Analysis Plan and perform independent statistical analysis of Program data.

1.1 RTI Field Study

To address the first goal, a field study was planned and conducted by RTI. In this study, identical twins and a family member (sibling or parent) were recruited. Each group went to either Williamsburg, VA, or Research Triangle Park, NC, for a four-day stay at a hotel. During this stay, daily sweat samples were collected onto polydimethylsiloxane membranes, as described in earlier reports to DARPA/ARO. A total of seven sets of twins were recruited. The goal was 30 twin pairs. Given the relatively poor response rate and the need for project resources to adequately address the data processing and statistical analysis needs of the overall USD program, the field study was terminated.

1.2 Data Processing and Statistical Analysis

An important aspect of the project centered on the most effective manner in which to process the data (mainly raw data from gas chromatography/mass spectrometric analysis) obtained from the study teams. As the work unfolded, the teams' different approaches as to handling the data became clear; this aspect was more complicated than had been anticipated. DARPA felt that there needed to be an effort in which the statistical analysis members of all of the teams communicated and defined issues as well as discussed options for approaching and solving the problems at hand. RTI coordinated that effort and, at the direction of DARPA, organized a working group that began with bi-weekly teleconferences and culminated in a workshop held at RTI International in Research Triangle Park, NC. The content and

recommendations that arose from that workshop were provided to ARO/DARPA in an earlier report.

This report describes the approach that RTI developed to permit the analyses needed to address the USD objectives set forth by DARPA. It describes the procedures devised for importing raw data, all steps of processing the data, and the different approaches to statistical analysis. This report also includes instructions for using the various programs involved. Finally, the report demonstrates the applicability of the approach via application to real data sets acquired by other USD team members.

2.0 Data Processing

2.1 Objectives

Providing an independent processing and analysis approach to data provided by the three external research groups was an underlying objective of this program. Through numerous discussions with these providers and with DARPA project management, this objective was interpreted to require that only data generated by gas chromatography/mass spectrometry (GC/MS) be processed by RTI. Data would be provided in raw form, devoid of any post-acquisition manipulation, either by the generating instrument or the attending instrument operator. A diversity of data formats created from software applications on multiple GC/MS systems would need to be accommodated. Furthermore, this data would be three-dimensional: consisting of time, intensity, and mass per unit charge (m/z) coordinates.

Because the significance of chemical concentration or mass—as reflected by chromatographic peak size—on the characterization of odor signature is unknown, no presumptions could be made regarding the inclusion, or exclusion, of peaks on the basis of size. In terms of preliminary data processing, very large peaks were afforded the same importance as very small peaks. Similarly, no assumptions were made regarding homogeneity, or heterogeneity, of individual chromatographic responses. Our extensive experience in separation science has shown that chromatographic resolution is always imperfect, particularly when challenged by highly complex chemical mixtures as would be expected in biological systems.

A significant advantage of our chromatographic data processing approach was realized through utilization of the m/z data dimension. Deconvolution of unresolved peaks, peak alignment and peak identification all utilized the uniqueness of the mass fragmentation pattern, in addition to the time dimension.

Overall, the data processing approach which we promoted is applicable to GC/MS data from any commercially available instrument and requires no intervention by the instrument operator other than to create the raw data output file.

2.2 Methods

2.2.1 Receiving and Importing Raw Data

At the provider's discretion, raw data were transmitted to a secure FTP server at RTI under individual, password-protected FTP accounts established for each provider. In addition to the raw data itself, relevant sample analysis parameters (e.g., sample type, extraction procedure, chromatographic conditions, internal standards, replicates, etc.) were requested to facilitate RTI's data processing and analysis. Raw data were moved from the FTP server to a project share and organized into folders based on information provided in the accompanying analysis parameters file. This share was accessible only to authorized individuals and was backed up nightly.

Prior to initiating formal data processing on a given data set, representative chromatograms were visually evaluated, using the particular application software in which they were created, to assess component complexity, baseline drift, detector noise, and anomalous m/z assignment resulting from improper instrument tuning. As needed, project staff were consulted to

determine the impact of these assessments and to modify the data processing scheme. If necessary, instrument-related problems were brought to the attention of appropriate staff from the contributing organization and remedial actions implemented to correct the problem.

Two GC/MS instrument manufacturer/data systems and, therefore, two data file formats were encountered on this program: Finnigan Xcalibur® and Agilent ChemStation®. Both of these data systems incorporated utilities for conversion to the generic netCDF format compatible with the subsequent peak deconvolution step. After the netCDF files were created, the raw data files were moved from the project share to archival storage. The archival storage location was backed up weekly.

2.2.2 Detecting Components

An investigation of relevant literature led to the discovery of a novel component detection method developed at the National Institute of Standards and Technology (NIST). The Automated Mass Spectral Deconvolution and Identification System (AMDIS, Version 2.6, 2004) program satisfied many of the criteria for data processing imposed by this project, including:

- It is compatible with many instrumental (Finnegan, HP, etc.) and netCDF file formats,
- Utilizes the m/z dimensionality of the data in addition to the chromatographic profile
- It retains spectral information for minor sample constituents
- It is configurable to allow adjustment for noise and chromatographic complexity
- Multiple possible components can be detected within a single peak
- Automatic baseline removal
- Customized mass spectral libraries can be created for specific sample/analyte types
- Capable of detecting MS instrumental tuning problems

AMDIS developers were invited to RTI early in the project to present an overview of program features and to provide guidance on setup and configuration. Preliminary assessments on actual odortype datasets were encouraging and provided valuable insights into program configuration settings which would yield optimal results. These setting were balanced so as to detect all components in the chromatogram without generating spurious, or “false positive” components. The nondefault settings that were used were:

- Deconvolution
  - Resolution: Low
  - Sensitivity: Very Low
- Identification
  - Analysis Type: Simple

Individual chromatographic runs were imported, individually, into AMDIS, either in the raw instrument file formats or, more typically, as netCDF files. Output text files were generated from each run to be compatible with subsequent data processing steps.

2.2.3 Smoothing

Component detection using AMDIS was especially sensitive to undersmoothed or oversmoothed data. When the data were undersmoothed, AMDIS found a high level of noise in the chromatograms. As a result, chromatographic peaks that appeared by visual inspection to be

components were not detected by AMDIS. When the data were oversmoothed, AMDIS found a very low noise level. In these cases, AMDIS detected components at very small perturbations in the chromatogram, even though visual inspection led us to believe that these were nothing more than noise in the data.

The literature describes many methods for smoothing. We spent a significant amount of time deciding which smoothing method was appropriate for our needs. We focused on methods that could be applied to each individual ion chromatogram (IIC), because AMDIS operates on that level, in addition to the total ion chromatogram (TIC). Poorly smoothed IICs cause just as many problems for AMDIS as do poorly smoothed TICs. Also, we eliminated methods, such as CODA (Windig et al., 1996) or the Morphological Score (Shen et al., 2001), that remove or penalize entire IICs. We found that although the IICs may be unsmooth, they still may provide important component information somewhere in the chromatogram. The methods we considered were:

- Fourier transform. We used the *fft* and *ifft* functions in SAS/IML (Version 8.02, 1999) and tested five different cut-off points: 50, 60, 70, 80, and 90. For each cutoff point P, we set the upper (100-P)% of the Fourier coefficients to 0.
- Savitzky-Golay (SG) filter. We used the *sgolayfilt* function as contributed to Octave (http://octave.sourceforge.net/index/f/sgolayfilt.html) and ported to Matlab (Release 11). For parameters, we used a polynomial order of 2 and tested window widths of 11, 17, and 21, as well as some earlier testing with window widths less than 9.
- Wavelets. We used the *call wavft* and *call wavift* routines in SAS/IML and tested 37 different combinations of parameters.
- Splines. We used the *call spline* routine in SAS/IML and tested smoothing parameters of 100 and 1000.

We applied the different smoothing methods to a few representative sample files, and we selected the methods that appeared to work the best for further study. For these best methods, we chose one representative sample file and let AMDIS detect components in the smoothed version of the file. We tried three different AMDIS parameter combinations. We evaluated the performance of each smoothing method by subjectively noting the level of false positives (AMDIS detected a component when it appeared to be noise) and false negatives (AMDIS failed to detect what appeared to be a component). We assigned levels of very low, low, moderate, high, and very high to the numbers of false positives and false negatives. Table 2-1 provides a summary of our results. The methods that displayed the best combination of false positives and false negatives were Spline(1000), SG(17), and SG(21), which each scored very low false positives and low false negatives. Each of the three performed approximately the same, so we used other criteria for making our final choice. SG(17) and SG(21) both did well using our default AMDIS settings, but because Spline(1000) required modified AMDIS settings, we eliminated it. We chose SG(17) over SG(21) as our smoothing method because it gave us similar performance with less modification of the data.

Table 2-1. Levels of False Positives and False Negatives for Various Smoothing Methods

Smoothing Method	AMDIS Settings	Detected Components	False Positives	False Negatives
Unsmoothed	M/M	182	High	Moderate
Unsmoothed	L/L	36	Moderate	High
Unsmoothed	L/VL	13	Very low	Very high
Wavelet (22)	M/M	114	High	Moderate
Wavelet (22)	L/VL	14	Very low	Very high
Wavelet (34)	M/M	132	High	Moderate
Wavelet (34)	L/L	40	Very low	High
Wavelet (34)	L/VL	20	Very low	Very high
Spline (100)	M/M	117	Moderate	Low
Spline (100)	L/L	60	Very low	Moderate
Spline (100)	L/VL	29	Very low	Very high
Spline (1000)	M/M	70	Very low	Low
Spline (1000)	L/L	42	Very low	High
Spline (1000)	L/VL	21	Very low	Very high
SG (11)	M/M	690	Very high	Very low
SG (11)	L/VL	45	Very low	Moderate
SG (17)	M/M	759	Very high	Very low
SG (17)	L/L	152	High	Very low
SG (17)	L/VL	77	Very low	Low
SG (21)	M/M	682	Very high	Very low
SG (21)	L/VL	86	Very low	Low

The AMDIS Settings column indicates the settings for Resolution and Sensitivity, where M=Medium, L=Low, and VL=Very Low.

Not all experiments required data smoothing. In fact, most did not because the data were already smooth enough in raw form to allow AMDIS to perform adequately. To decide whether to smooth the data, we first performed our data processing steps as if we were not smoothing. After running AMDIS, if the number of detected components did not match our expectations based on visual review of the files, we inspected the files more closely and decided whether smoothing would be beneficial. If we did smooth the data, we verified that it was indeed beneficial after running AMDIS on the smoothed data.

The application of the selected smoothing method required several steps. The CDF file containing the full matrix of intensity values at every retention time and m/z value was converted to Matlab format using the NetCDF toolbox for Matlab (http://woodshole.er.usgs.gov/staffpages/cdenham/public_html/MexCDF/nc4ml5.html). The SG smoothing algorithm was then applied in Matlab. Finally, the smoothed Matlab dataset was converted back to CDF format, again using the NetCDF toolbox. The resulting CDF file was input to AMDIS.

2.2.4 Clustering Components

One requirement of our data analysis methods was that we could construct a list of sample files in which each detected compound appeared. Because AMDIS detected components separately in each file without knowledge of components in the other files, we had to find a way

to connect components representing the same compound together across files. We decided to use a clustering approach to make these connections. In this approach, components with sufficiently similar retention times and mass spectra would be placed into the same cluster. The cluster would later be linked to some chemical compound, and all the files represented in that cluster would be the files in which the compound appeared, thus satisfying our data analysis requirement.

Because retention time was one of the clustering dimensions, we had to guard against situations where chromatograms were poorly aligned, which would cause components representing the same compound to have different retention times. If the difference were large enough, the components would be placed into separate clusters. To help avoid these errors, we applied a coarse alignment algorithm to the files before performing any clustering. This algorithm adjusted the retention times of all components so that each internal standard and other commonly detected component had identical retention times across all files.

2.2.4.1 Finding Internal Standards and Landmarks

The first step in the coarse alignment algorithm was to find internal standards and other commonly detected components that we called "landmarks." The data providers supplied the identities of the internal standards and their expected retention times. Based on the identity, we were able to look up the expected mass spectra of the internal standards. We chose candidates for landmarks through visual inspection of the chromatograms. AMDIS output was used to determine their expected retention times and mass spectra.

Armed with the expected retention times and mass spectra, internal standards and landmarks could be identified manually in every file. However, given the large number of files, we had to automate the process to make it feasible. We implemented a searching algorithm to identify which of the AMDIS-detected components in each file were most likely to be the components of interest. After completing the search, we reviewed the assignments to check for any clues that might indicate a missed assignment and modified the assignments manually as necessary.

To run the search procedure for a target component (either an internal standard or landmark), the user supplied the expected retention time, the expected mass spectrum, and a retention time window. The expected mass spectrum was limited to the three to five mass values with the highest intensities. We found that this was enough to make a good identification, and more mass values tended to cause too many incorrect assignments. The search was restricted only to components detected within the retention time window. This was used to prevent the algorithm from making false assignments to components that were so far away from the expected retention time that they could not have possibly come from the same compound. The size of an appropriate retention time window depended on how well-aligned the data seemed to be already. We generally started with a window radius of 0.5 minutes and modified it if necessary.

For every component in the retention time window, the algorithm calculated the Euclidean distance between the detected component's AMDIS-produced mass spectrum and the expected mass spectrum. The calculation was limited only to those mass values included in the expected mass spectrum. Components were disqualified if either of the following conditions was true:

- Any mass value in the expected mass spectrum was absent from the component's mass spectrum.
- The mass value with the highest intensity value in the component's mass spectrum was absent from the expected mass spectrum.

Among the remaining components in the retention time window, the one with the smallest distance was assigned as the match to the target component.

When reviewing the assignments, we looked for several indicators that an incorrect assignment was made for a particular file:

- The distance between the expected mass spectrum and the mass spectrum for the assigned component was large relative to the rest of the files.
- The distance between the expected retention time and the retention time for the assigned component was large relative to the rest of the files.
- No assignment was made. This meant either that no component was detected within the retention time window or that all detected components were disqualified.

If an assignment looked suspicious, we manually inspected the file to determine whether the assignment was indeed incorrect and, if so, how to correct it. Possible corrective measures included adding or removing mass values from the expected mass spectrum, increasing or decreasing the size of the retention time window, manually making the correct assignment, or discarding the file from analysis. The first two measures were taken when there was more of a global problem, such that the same modification would fix several incorrect assignments. Making assignments manually was performed when only one or a few files had a problem. Discarding a file from analysis was necessary when our inspection of it led us to believe that it was of low chromatographic quality.

When deciding how many landmarks to include, we took into consideration the number of internal standards and the quality of the alignment before any processing. If the set of files were well aligned already, we felt that a total of three standards and landmarks would be sufficient. We tried to get one in the early part of the chromatogram, one in the middle part, and one in the later part. When files were not well aligned, we increased the number of landmarks, but not to more than about five or six.

When choosing which components to use as landmarks, we looked for components that were distributed throughout the retention time span and for components that appeared in most of the files. We found that a good starting point was to look in blank sample files for landmark candidates. We started with the blanks because any compounds that appeared in the blanks should have appeared in all the true sample files, also. If the blanks did not contain enough landmark candidates, then we randomly chose other sample files in which to look for candidates. After identifying the candidates, we used the search algorithm described above to see how many files contained the candidate. If this number was sufficiently large (approximately 75%), then we accepted the candidate as a landmark.

2.2.4.2 Coarse Alignment

Once all the internal standards and landmarks had been found, we applied a coarse alignment algorithm to ensure that the chromatograms were at least somewhat aligned. The coarse alignment algorithm had two steps:

1. It adjusted the retention times of internal standards and landmarks so that the new retention times were identical across all files for each standard or landmark. The target retention time for each standard or landmark was the average of the unadjusted retention times across all files.
2. It adjusted the retention times of all other components so that the ratio of the following two distances remained the same before and after the coarse alignment:
    a. the distance from the component to the nearest standard or landmark earlier in the run, and
    b. the distance from the component to the nearest standard or landmark later in the run.

All retention time adjustments were performed multiplicatively, not additively. That is, all adjustments represented a stretch or compression of the retention time axis, not a shift of the axis.

2.2.4.3 Multidimensional Clustering

After the coarse alignment, we applied a clustering algorithm to collect components from different files into groups likely to represent the same chemical compound. We used the SAS/STAT (Version 8.02, 1999) procedure *fastclus* to perform the cluster analysis. We set the procedure to perform a disjoint cluster analysis using nearest centroid sorting, and we used the retention time and mass spectrum as the clustering dimensions. The inclusion of both retention time and mass spectrum as clustering dimensions was very important to the success of the clustering algorithm. We found that ignoring either would result in numerous errors, generally from placing components into the same cluster when they actually represented different chemical compounds.

Ignoring retention time caused problems because components with very similar mass spectra were often detected at very different retention times. Large differences in retention times in otherwise well-aligned chromatograms imply that the components must represent different chemical compounds. Had we clustered on mass spectra alone, though, such components would likely have been placed into the same cluster. Including the retention time helped ensure that these components were split into separate clusters.

Ignoring the mass spectra caused problems because components representing different compounds often had overlapping elution profiles. Therefore their retention times were very similar. If we were clustering only on retention time, these overlapping components would most likely be placed into the same cluster. Including the mass spectra helped ensure that these components were split into separate clusters.

All detected components from all files were included in the cluster analysis. Each component was characterized by a vector that included its adjusted retention time, in minutes,

and the intensity at each mass value in its mass spectrum. The mass intensities had been scaled by AMDIS so that the largest for each component was set to 999. If the data contained m/z values in the range 40 to 400, then the vector was of length 362 (1 retention time value and 361 m/z values). The vector was used to calculate distances in the clustering algorithm. Because the vector consisted almost entirely of mass spectrum values and because the scale of the retention times was smaller than the scale of the intensities, the distance between components was dominated by the distance between mass spectra. However, we believed retention time was a very important factor and should have more of a contribution to the distance formula. To achieve this, we applied a weight factor to the retention time dimension before calculating the distances.

Two parameters were supplied to the clustering algorithm: the retention time weight and the maximum cluster radius. We experimented with different parameter values to find a reasonable combination, and we settled on a retention time weight of 5000 and a cluster radius of 1500. After running the clustering procedure with these parameters, each detected component was assigned to a cluster. The number of clusters varied quite a bit from one experiment to the next, from a few hundred to several thousand.

Because of imperfect component detection and spectrum deconvolution by AMDIS and the nature of the clustering algorithm, some clustering errors occurred. Cluster errors included having components from the same chemical compound present in more than one cluster and having components from different chemical compounds present in the same cluster. Effort was made in later steps when flagging heterogeneous clusters (section 2.b.vii.) and performing statistical dimension reduction (section 3.b.) to detect and handle some of these errors.

2.2.5 Finding Core Clusters

Every sample file contained many compounds originating from numerous sources, including diet, environment, the sample workup process, and genetics. We were interested only in compounds that were genetic in origin. For this reason, we took several steps to remove or at least flag compounds we suspected were not genetic. The first step was to identify "core" clusters, or those that had consistent representation among files from at least one genetic type. We reasoned that if a compound is derived from a subject's genetics, then that compound should show up in every sample file. Of course, because of instrument detection limits, noisy data, and other abnormalities, a truly genetic compound has some probability of not being detected in a given sample file, and so we would expect a detection rate somewhat less than 100% even for a genetic compound. The expected rate depends on the probability of detection. A compound with a detection rate significantly below the expected rate is likely not a genetic compound. Because we did not know the probability of detection, we experimented with different expected rates and chose the one that gave the most reasonable results while still making sense in the context of the experimental design.

Detection rates were set at each stage of the experimental design. For example, suppose the design was set up so that for each genetic type there were 10 donors, each donor donated 3 samples, and each sample was subjected to 5 replicate analyses. We set a minimum detection rate at the replicate level, at the sample level, and at the donor level. At the replicate level, our minimum detection rate may have been 60%, meaning a cluster had to be represented in at least 60% (3/5) of the replicates for a particular sample to be classified as core for that sample. At the sample level, our minimum detection rate may have been 100%, meaning a cluster had to be core

in all samples for a particular donor to be classified as core for that donor. Finally, at the donor level, our minimum detection rate may have been 70%, meaning that a cluster had to be core for at least 70% (7/10) of the donors from a particular genetic type to be classified as core for that genetic type. Clusters classified as core for at least one genetic type were designated as core clusters.

In the example above, we set detection rates to a constant percentage at each level. We used this method when the design was balanced (i.e., when each genetic type had the same number of donors, each donor donated the same number of samples, etc.). In unbalanced designs, we believed that using a constant percentage may not be appropriate. Being detected in 50% of the files may be very different when the 50% is 1 out of 2 compared to when the 50% is 10 out of 20. It is very dependent on the probability of detection. In these cases, we used a hypothesis testing approach to decide whether or not a cluster was core at the unbalanced level. Suppose we are at the sample level. Our null hypothesis was that the compound represented by the cluster was not present for that donor. Under the null hypothesis, the number of samples in which the cluster was present followed a binomial distribution with n equal to the number of samples and p equal to the probability of a false positive, that is, the probability that a compound was present in a sample given that it was not present in the donor. Using an alpha level of 0.05, we calculated the minimum number of samples in which the cluster must be present to reject the null hypothesis and conclude that it was indeed present in the donor. We set the probability of a false positive, and, because we did not know the true value of this probability, we experimented with different values to obtain a manageable number of clusters (approximately 100).

Once we had classified all clusters as either core or not core, we kept only the core clusters for further analysis.

2.2.6 Searching the NIST Library

Chemical identities were assigned to each component using the NIST MS Search Program for the NIST/EPA/NIH Mass Spectral Library (V2.0). Text files containing individual mass spectra were created in a previous step and were imported directly into the MS Search program. The individual mass spectra were matched against the library and the best two library “fits” output to a SAS program which compiled all results into a single dataset.

To assess the accuracy of this automated approach, independent compound identifications were made, on a small subset of spectra, by an expert in mass spectral interpretation. Out of 44 total identifications, the expert completely agreed with the automated identification in 25 cases (57%); an additional 11 identifications (25%) by the NIST program were found to be on heterogeneous components; these components had a primary chemical compound, which was correctly identified, and at least one other compound of lesser amount. The expert deemed the remaining 8 (18%) identifications by the NIST program to be incorrect and determined that the most probable reason for the incorrect assignment was multiple chemical species contributing to a single mass spectrum. Overall, the NIST program was able to determine the principal chemical species responsible for the mass spectrum in 82% of cases.

2.2.7 Flagging Heterogeneous Clusters

Given a likely identity for each detected component, we were able to assign identities to each core cluster. Because a cluster consisted of many components that were grouped together based on their similar characteristics, we expected most of the components in a cluster to have the same identity. When this was true, we assigned that common identity to the cluster. When it was not true, however, it indicated that there may have been an error in forming the cluster, and so we flagged it as heterogeneous. One such error could be that the cluster contained components that were derived from several different compounds instead of just one. Later analyses were performed with and without heterogeneous clusters. These clusters were not completely excluded from later analyses because it was possible that the components were all derived from the same compound, but that compound just was not in the NIST library. In this case we may expect the results to be scattered over several different compounds that did happen to be present in the library.

To decide whether a cluster was heterogeneous, we first collected the NIST library search results for all components in that cluster. We included the top two hits for each component. Later investigation revealed that although the results were largely the same for our cases, using the top one hit instead of top two hits is a slightly better method. Once we had all the search results, we computed a weighted frequency distribution of the hits, where the match factor from the search result was used as the weight. Using this weight value gave the most emphasis to matches considered to be the best. If the highest frequency library hit had a frequency less than 60%, then the cluster was flagged as heterogeneous.

2.2.8 Quantification

Our statistical analysis methods required some measure of the "quantity" of each compound in each file. We used the area under the compound's elution profile as the base of our quantification. We then normalized this area to the area of the internal standard(s) to account for varying levels of injection amount in each sample. Finally, we applied a log transformation to the normalized area.

We chose to use the area of the internal standard instead of the area of the entire chromatogram as our normalization factor because we believed the area of the entire chromatogram could be heavily influenced by the presence or absence of environmental compounds. Therefore there is no reason to believe that the area of the entire chromatogram should be equal across sample files. Conversely, the area of the internal standards should be the same across sample files, and so it is a more appropriate normalization factor.

The area under the compound's elution profile was provided indirectly by AMDIS. The area produced by AMDIS had been normalized to the area of the entire chromatogram. Because we wanted to use another normalization factor, we first un-normalized the AMDIS-produced area. We then divided the areas of all components in a sample file by the area of the internal standard for that file. In experiments with multiple internal standards, we used the mean of the internal standard areas as the normalization factor. In experiments with no internal standards, we did not perform normalization.

We next replaced any areas equal to 0 with some small nonzero area. An area was equal to 0 if a compound was not detected in a particular file. We wanted to be conservative by assuming that the compound was present but was below the detection limit. We replaced the 0 areas with 1/10 of the smallest detected area for the experiment.

Finally, we applied a natural log transformation to the data to make the data more symmetrical in distribution and to help stabilize the variance. These two data characteristics were assumptions of some of the analysis methods.

3.0 Dimension Reduction

3.1 Biochemical

Among the identifications assigned to each cluster were chemical species which are highly improbable in biological systems. These identifications were classified as “impossible” and included:

- siloxanes and other silicon-containing species
- halogenated compounds (F, Cl, Br, I)
- phthalates and compounds containing benzenedicarboxylic acid moieties

Siloxanes are commonly associated with treated glass surfaces and are, therefore, almost certainly artifacts of sample handling or chromatographic analysis. Although halide ions, notably chloride ion, are very common in the human body as important electrolytic contributors, halogenated compounds are extremely rare and, if found, are generally associated with exposure to a hazardous chemical (e.g., carbon tetrachloride). Finally, phthalates are included on the “impossible” list because of their ubiquitous presence in the environment. They are intermediate chemicals in the formulations of common plastics and, as such, are pervasive in the environment. Identifications which included any of these chemical species were retained in the dataset and were flagged with an “I.”

In addition to the impossible compounds, an additional list of “environmental” chemical species was compiled from the cumulative findings of Draper Laboratories and Monell Chemical Senses Institute. These environmental compound types included:

- Food antioxidants
- Tobacco components
- Coffee components
- Fragrances and flavors (food additives)
- Industrial Pollutants
- Insecticides/Pesticides
- Medications
- Plastics/plasticizers
- Volatile Organic Compounds

As in the case of the “impossible” compounds, these identifications were retained in the dataset and flagged with an “E.” In addition to their anthropogenic sources, many of these “environmental” chemicals (e.g., selected ketones) could also emanate from human biological sources. Consequently, subsequent statistical analyses were performed both with and without these identifications in the data.

3.2 Statistical

Two statistical procedures are used to further remove the chemical compounds that do not provide significant help in distinguishing the genotypes. These two methods—analysis of variance and stepwise linear discriminant analysis—are described below.

3.2.1 ANOVA

Compounds whose concentration levels are significantly different for different genetic types are likely to be important in distinguishing those genetic types. Conversely, compounds whose concentration levels are essentially the same from one genetic type to the next are likely to be unimportant in distinguishing the genetic types. One way to identify whether a compound's concentration level differs significantly between genetic types is to use ANOVA. For each experiment, we performed an ANOVA on each compound independently and used the result to judge whether the compound significantly distinguished the genetic types in that experiment. This resulted in a smaller set of compounds known as the ANOVA-reduced set.

The following is the general ANOVA model we used for each compound:

$$Y_{ijklm} = g_i + d_{ij} + t_{ijk} + s_{ijkl} + \varepsilon_{ijklm}$$

where:

Y_{ijklm}	=	observed quantity of the compound for replicate *m* from sample *l* from donation day *k* from donor *j* from genetic type *i*
g_i	=	effect of genetic type *i*
d_{ij}	=	effect of donor *j* from genetic type *i*
t_{ijk}	=	effect of donation day *k* from donor *j* from genetic type *i*
s_{ijkl}	=	effect of sample *l* from donation day *k* from donor *j* from genetic type *i*
ε_{ijklm}	=	random error

Each effect was nested within the preceding effect in the model, and all effects except the genetic type effect were random effects. For experiments that did not include all levels of the model, the inapplicable levels were dropped. The ANOVA models were run using the SAS/STAT (Version 8.02, 1999) procedure *glm*. Compounds were considered significant if the *p*-value of the overall *F* test was below an alpha level. The alpha level was initially set to 0.05, but was moved up or down if the number of signification compounds was deemed too low or too high.

3.2.2 Stepwise Linear Discriminant Analysis (SLDA)

The problem of linking chemical profile of a sample uniquely to a genotype can be cast in a classification framework. Just as it is possible to match a fingerprint to the fingerprints with known sources in a database, one can match a chemical profile to the chemical profiles with known sources in a database. Unlike a fingerprint, however, the chemical profile of a genotype changes significantly with environmental factors, and all the chemical profiles produced under different conditions form a distribution, which requires a group of chemical profiles to characterize the distributions. Assuming that each genotype corresponds to a unique distribution of chemical profiles, the problem becomes how to assign an observed chemical profile to the genotype associated with it. This is therefore a classification problem, and we expect that a variable selection procedure based on a classification method should be suitable for selecting the best classification variables. We applied Stepwise Linear Discriminant Analysis (SLDA) for variable selection.

For a set of observations containing one or more quantitative variables and a classification variable defining groups of observations, a classification rule develops a discriminant criterion to classify each observation into one of the groups. The derived discriminant criterion can be used to classify a new observation with unknown group membership. Linear discriminant analysis (LDA) develops a discriminant criterion that is linear in the quantitative variables. The original linear discriminant analysis method applied to two-group problems. It can be generalized to handle the many-group case. In this general approach, one assumes that each group has a multivariate normal distribution, and a classification criterion using a measure of generalized squared distance is developed. The classification criterion is based on the pooled covariance matrix (or the individual within-group covariance matrix). Each observation is placed in the group from which it has the smallest generalized squared distance.

SLDA is used to select a subset of the quantitative variables for use in discriminating among the groups. This analysis is a useful prelude to further disciminant analyses.

Stepwise selection is very similar to that of stepwise regression. It begins with no variables in the model. At each step, the model is examined. If the variable in the model that contributes least to the discriminatory power of the model as measured by Wilks' lambda fails to meet the criterion to stay, then that variable is removed. Otherwise, the variable not in the model that contributes most to the discriminatory power of the model is entered. When all variables in the model meet the criterion to stay and none of the other variables meets the criterion to enter, the stepwise selection process stops.

Compared with ANOVA, SLDA is expected to have two advantages:

1. It takes into account the correlations among classification variables, and
2. It tries to select the variables with the most discriminatory power.

When we applied SLDA to the simulated as well as real data, we used the intensities of the chemical compounds as classification variables and genotype as the classification label. We selected a subset of the chemical compounds.

4.0 Statistical Methods for Classification

4.1 Linear Methods

4.1.1 Linear Discriminant Analysis (LDA)

This method is discussed in the above section. This is the simplest method for solving the classification problem. The advantage of this method is its simplicity and effectiveness. It is a very efficient method when the intensities approximately follow a multivariate normal distribution. It may not perform well when the genotypes can not be linearly separated, or the data distribution drastically differs from multivariate normal distribution. When applied to the real data, the intensities of the selected chemical compounds were used to classify the genotypes of the samples.

4.1.2 CDA

Canonical discriminant analysis is a dimension-reduction technique related to principal component analysis and canonical correlation. In a canonical discriminant analysis, linear combinations of the quantitative variables that provide maximal separation between the groups are constructed. Given a classification variable and several quantitative variables, canonical discriminant analysis derives *canonical variables*, linear combinations of the quantitative variables that summarize between-class variation in much the same way that principal components summarize total variation.

Given two or more groups of observations with measurements on several quantitative variables, canonical discriminant analysis derives a linear combination of the variables that has the highest possible multiple correlation with the groups. This maximal multiple correlation is called the *first canonical correlation*. The coefficients of the linear combination are the *canonical coefficients* or *canonical weights*. The variable defined by the linear combination is the *first canonical variable* or *canonical component*. The second canonical correlation is obtained by finding the linear combination uncorrelated with the first canonical variable that has the highest possible multiple correlation with the groups. The process of extracting canonical variables can be repeated until the number of canonical variables equals the number of original variables or the number of groups minus one, whichever is smaller. Thus, dimension reduction is achieved. The classification can be performed on the canonical variables.

Compared with LDA, CDA offers the function of dimension reduction. This is useful when the number of chemical compounds and the number of genotypes are large. On the other hand, the dimension reduction may reduce the effectiveness of the classification algorithm. In our simulated and real data sets, the number of genotypes was not terribly large and the number of chemical compounds was successfully reduced to a manageable amount. Thus the advantage of CDA was not obvious. However, in the future scenarios where the number of genotypes is large and further dimension reduction becomes important, this method would be worth investigating. Because this is a linear classification method, it would not be effective if the genotypes cannot be separated linearly.

4.2 Nonlinear Methods

We investigated two nonlinear classification methods— Support Vector Machines (SVM) and Generalized Discriminant Analysis (GDA)—in case the data contained some nonlinearity that could be useful to help separate the genotypes and would adversely impact the performance of the linear methods.

4.2.1 SVM

SVM are algorithms that can perform binary and multi-group classification and real valued function approximation (regression estimation) tasks. They are a family of learning algorithms that is considered one of the most efficient methods in many real-world applications.

For a two-group classification problem where the two groups can be perfectly separated by a linear function (i.e., a hyperplane), SVM finds the hyperplane that separates the two groups with the largest margin, where margin is the smallest distance of a data point to the hyperplane. When the two groups can not be separated by a hyperplane, SVM can transform the data into a high dimensional space such that in this new space the groups can be perfectly separated. SVM can then treat the problem as a linear problem in the new space. Some versions of SVM do not require the groups being perfect separated. Instead, they try to find the hyperplane that minimizes the classification error. SVM have been generalized to classify many-group case. SVM use something called kernel to transform the data into the high dimensional space. Using kernel makes the transformation implicit and the computation efficient. Many different kernels have been developed, and each of them corresponds to a particular nonlinear transformation. The most popular kernel is the Gaussian kernel, which has been proven to be efficient and adequate by many applications. We used the Gaussian kernel for the current study.

It is not clear that if the genotypes can be linearly separated or not. However, given the relative small sample size, the data points in the high dimensional space of the intensities are very sparse. Chances are that any nonlinearity in the data will not be adequately captured by the data. So the advantage of the SVM is not realized by small samples and LDA may be adequate. This was demonstrated by the application to the real data. In future analysis where the sample size is significantly larger, SVM is expected to be a very competitive method. It may capture some nonlinearity in the data which will be missed by linear methods.

4.2.2 Generalized Discriminant Analysis (GDA)

The Generalized Discriminant Analysis (GDA) is a nonlinear extension of the ordinary Linear Discriminant Analysis (LDA). Using kernel functions, the data are mapped nonlinearly to a high dimensional feature space with linear properties, similar to the case of SVM. In the new feature space, the classical LDA is applied. As with SVM, using different kernels, a wide class of nonlinearities are covered. Some literature (Baudat and Anouar, 2000) suggested that GDA had similar performance as SVM, based on limited simulations. However, GDA is not as well studied or widely applied as SVM. One advantage of SVM is that it does not assume normality of the data in the feature space.

5.0 Application to Simulated Data

Our primary objective here was to compare our analysis methods. We hoped that, through the use of standard data, the benefits of the different approaches could be evaluated while concurrently creating a systematic walk-through of the approaches to help illustrate the logic for instructional purposes. We sought to answer the following questions to help us meet our primary objective:

1. How often do the variable selection methods choose the right set of compounds? How often are there false positives? False negatives?
2. What are the classification rates for each classification method?
3. Do the classification rates differ depending on which variable selection method is used?

We used separate training and testing datasets when analyzing the simulated data to answer these questions.

A secondary objective was to measure cross-validation bias. Real-world situations generally do not have enough data to use separate training and testing datasets, and one alternative is to use cross-validation instead. Many believe that cross-validation tends to increase the classification rate above what would have been observed had separate training and testing datasets been used. This difference is known as the cross-validation bias. To measure cross-validation bias, we calculated the classification rate using cross-validation for some of the training datasets. We then compared the distributions of these rates to those obtained using separate testing datasets.

Appendix A provides a set of instructions for how the various components are used. Appendix B provides a list of necessary software. Software was delivered to DARPA via uploading to the USD web site.

5.1 Data Specifications

5.1.1 Linear Datasets

Each simulated dataset consisted of randomly generated, independent observations from a multivariate normal distribution with a given mean and covariance structure. An observation contained intensity values for several hypothetical compounds. These intensity values represented GC/MS analysis data after it had undergone our data processing steps. Different genotypes were simulated by generating data using different means in the multivariate normal distribution.

We generated simulated datasets using several scenarios. We compared the selection and classification methods in the presence of a low number of genotypes (3) and a higher number of genotypes (10), in the presence of low genotype separation and high genotype separation, and in the presence of low correlation between compounds (ρ=0.2) and high correlation (ρ=0.8). Table 5-1 provides a summary of the scenarios considered in the simulation.

Table 5-1. Summary of Linear Simulation Scenarios

Scenario	Number of Genotypes	Genotype Separation	Correlation Between Compounds
1	Low	Low	Low
2	Low	Low	High
3	Low	High	Low
4	Low	High	High
5	High	Low	Low
6	High	Low	High
7	High	High	Low
8	High	High	High

An observation Y_i from genotype i was generated as:

$$Y_i = (y_{i1}, ..., y_{iC}) \sim N(\mu_i, \Sigma)$$

where

$(y_{i1}, ..., y_{iC})$ = the vector of intensity values in the observation
C = the number of compounds in the model
μ_i = the mean intensity vector for genotype i
Σ = the covariance matrix

For training datasets, 20 observations were generated for each genotype. This value was chosen because it is typical of what we have seen in the real data we have analyzed. For testing datasets, 200 observations were generated for each genotype.

We chose to use five compounds (C=5) in all our models. We designated the first three compounds as those that would separate the genotypes. We decided on three because it was the most we could have and still visualize the separation graphically. We added another compound that would not distinguish the genotypes, but would be correlated with the first three compounds. Finally, we added a fifth compound that would neither distinguish the genotypes nor be correlated with any of the other compounds. These last two compounds add some noise factors into the model and allow us to test how well our variable selection methods work.

We used a geometric approach to choosing the mean intensity vectors. The approach was slightly different in the 3-genotype model and the 10-genotype model. In both cases, though, the vector was specified as:

$$\mu_i = (\mu_{i1}, \mu_{i2}, \mu_{i3}, 0, 0)$$

where μ_{ij} is the mean of compound j in genotype i. We set the means for compounds 4 and 5 to a constant across all genotypes to force those two compounds not to contribute to the separation of the genotypes. We set the means for the remaining compounds geometrically. Each dimension in 3-dimensional space represents a compound. The mean of each genotype, then, can be represented as a point in 3-dimensional space. For the 3-genotype model, we envisioned the

genotypes as equally spaced points on a sphere centered at the origin. By controlling the radius of the sphere, we could control the distance between the genotypes. For the low separation scenarios, we set the radius to 0.5, and for the high separation scenarios, we set the radius to 1.5. On the unit sphere, the locations of the genotype means in the 3-genotype model are given in Table 5-2.

Table 5-2. Locations of Genotype Means in the 3-Genotype Model

Genotype	Mean Location
1	(0.707, 0.500, -0.500)
2	(0.259, -0.683, 0.683)
3	(-0.966, 0.183, -0.183)

For the 10-genotype model, we envisioned the genotypes as points on the 6 faces and 4 of the corners of a cube centered at the origin. By controlling the size of the cube, we could control the distance between the genotypes. For the low separation scenarios, we set the distance from the origin to a face of the cube to 0.3, and for the high separation scenarios, we set this distance to 0.5. On the cube where the distance from the origin to a face is 1, the locations of the genotype means in the 10-genotype model are given in Table 5-3.

Table 5-3. Locations of Genotype Means in the 10-Genotype Model

Genotype	Mean Location
1	(0, 0, 1)
2	(1, 0, 0)
3	(0, 0, -1)
4	(-1, 0, 0)
5	(0, 1, 0)
6	(0, -1, 0)
7	(-1, 1, 1)
8	(1, 1, -1)
9	(-1, -1, 1)
10	(1, -1, -1)

The covariance matrix is constructed as:

$$\Sigma = \begin{bmatrix} 1 & \rho & \rho & \rho & 0 \\ \rho & 1 & \rho & \rho & 0 \\ \rho & \rho & 1 & \rho & 0 \\ \rho & \rho & \rho & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where ρ is the pairwise covariance between compounds. In the low correlation scenarios we set ρ to 0.2, and in the high correlation scenarios we set ρ to 0.8. We set all the pairwise covariances equal because we wanted to limit the number of input parameters, limiting the number of confounding factors. Even though we expect real data to exhibit negative correlations as well as positive correlations, we elected to look at positive correlations only in this simulation. We

assigned the same covariance matrix to all genotypes for simplicity and because we see no reason to believe the covariance structure would differ from one genotype to the next.

Note that when constructing the covariance matrix, we assigned a value of 1 along the diagonal elements. This means the within-genotype variance for each compound is forced to be 1. In addition, it means the covariance matrix may also be thought of as a correlation matrix.

Genotype separation is characterized by the ratio of the between-genotype variance to the within-genotype variance. The within-genotype variance for every compound was set to 1. The between-genotype variance for a compound *j*, which is heavily influenced by the distance between the genotype means, is given by the formula:

$$\sigma_B^{\,2} = \frac{\sum_{i=1}^{k} n_i (\mu_{ij} - \overline{\mu}_i)^2}{k-1}$$

where:

$\sigma_B^{\,2}$	=	between-genotype variance
k	=	number of genotypes
n_i	=	sample size in genotype *i*
μ_{ij}	=	the mean of compound *j* in genotype *i*
$\overline{\mu}_i$	=	the mean of the mean vector in genotype *i*

Table 5-4 summarizes the genotype separation for the low separation and high separation scenarios.

Table 5-4. Summary of Genotype Separation

Number of Genotypes	Genotype Separation	Between/Within Variances	Pairwise Distances
3	Low	1.9 – 3.8	0.9
3	High	16.9 – 33.8	2.6
10	Low	1.2	0.4 - 1.0
10	High	3.3	0.7 – 1.7

5.1.2 Nonlinear Datasets

Each simulated dataset consisted of randomly generated, independent observations. Like the linear data, an observation contained intensity values for several hypothetical compounds. These intensity values represented GC/MS analysis data after it had undergone our data processing steps.

We used three schemes for generating nonlinear datasets. For each scheme, datasets were created using different scenarios, or combinations of parameters. In all scenarios, the number of hypothetical compounds was set to 3 to ease visualization of the data in 3-dimensional space.

Because the number of compounds was so small, we did not apply our variable selection methods to the nonlinear data. Instead, we focused only on comparing the classification methods.

For training datasets, 20 observations were generated for each genotype we included in the model. This value was chosen because it is typical of what we have seen in the real data we have analyzed. For testing datasets, 200 observations were generated for each genotype.

Scheme 1. The intensity values for the first two compounds were simulated with random values from a standard normal distribution. The intensity values for the third compound was set to the sum of the squares of the first two compounds, plus random noise generated from a standard normal distribution. In summary, each observation Y was generated as:

$$Y = (y_1, y_2, y_3),$$

where:

$y_1 \sim N(0, 1)$
$y_2 \sim N(0, 1)$
$r \sim N(0, 1)$
$y_3 = y_1^2 + y_2^2 + r + C_i$

In addition, to simulate different genotypes, an offset value was added to y_3. The actual value was dependent on the genotype i, and is depicted in the above specification as C_i. The magnitude of the offset affected the distance between each genotype and was varied in different scenarios, as shown in Table 5-5.

Table 5-5. Summary of Scenarios Under Nonlinear Scheme 1.

Scenario	Number of Genotypes	Offset Value		
		Genotype 1	Genotype 2	Genotype 3
9	3	1	2	3
10	3	2	4	6
11	3	3	6	9

Visually, data for a single genotype created under this scheme looked like a bowl or a vase. When all genotypes were combined together, data created under this scheme looked like a stack of bowls or vases.

Scheme 2. Data were created under Scheme 2 in much the same way as in Scheme 1. In fact, the only difference was that the random value assigned to y_1 was forced to be greater than 0. This requirement was implemented by discarding observations whenever y_1 was less than or equal to 0, and generating more observations until the desired number was met. Visually, data created under this scheme looked like a stack of bowls or vases that have been sliced vertically down the center. A summary of the scenarios created under Scheme 2 is presented in Table 5-6.

Table 5-6. Summary of Scenarios Under Nonlinear Scheme 2

Scenario	Number of Genotypes	Offset Value		
		Genotype 1	Genotype 2	Genotype 3
12	3	2	4	6

Scheme 3. A nonlinear transformation was applied to data from one of the simulated linear datasets. See the linear data specifications section for details on how the linear data was simulated. A nonlinear observation Y = (y_1, y_2, y_3) was created from a linear observation X = (x_1, x_2, x_3) by the transformation:

$$y_1 = x_1$$
$$y_2 = x_2^2$$
$$y_3 = x_3^3$$

A summary of the scenarios created under Scheme 3 is presented in Table 5-7.

Table 5-7. Summary of Scenarios Under Nonlinear Scheme 3

Scenario	Number of Genotypes	Source Linear Scenario
13	3	2
14	3	4
15	10	8

5.2 Comparison of Results

Results are based on 200 simulated training datasets and 1 simulated testing dataset. We applied the variable selection methods to each training dataset, and used the selected variables from the training datasets to train the various classification methods. We then used the testing dataset to calculate the classification rates for each method. Finally, we computed classification rates using cross-validation on 20 of the training datasets.

5.2.1 Assessing Variable Selection Methods

The first question we examined is the effectiveness of the variable selection methods. As stated previously, two methods were used in reducing the dimension of the datasets. One is a univariate approach based on the ANOVA method. The other is the Stepwise Linear Discriminant Analysis which takes into account the relationships (or correlations among the variables). Table 5-8 shows how well the two variable selection methods did in the linear simulations. Recall that in order to test the variable selection procedures, among the five components created in the data, the first three would contribute to separating the genotypes and the last two would not. A variable selection method performed "Exactly Right" if it picked the first three components and not the last two. A method produced "False Positive" only if it picked the first three components and one or more of the last two. A method produced a "False Positive and False Negative" if it missed at least one of the first three components and also picked at least one of the last two components. The desired situations are that the method produced either an "Exactly Right" result or a "False Positive" result, because our main purpose is not to include all important chemicals in data analysis and not to exclude them during the data reduction stage. Table entries indicate the percentage times the various types of results a method produced.

As indicated by table, in general, SLDA outperformed the ANOVA method, especially in the scenarios where the genotype separation is low (scenarios 1, 2, 5, and 6). It is not a surprise given that it uses more information in deciding which variables to retain. Another observation is that the number of false positives for SLDA tends to jump up for the scenarios with high correlation (scenarios 2, 4, 6, and 8). We think this is because component #4 is correlated with the first three, and SLDA is picking up the correlation structure between #4 and the first three components and therefore leading to the higher rate of false positives.

Table 5-8. Percentages of Times Variable Selection Methods Pick the Correct Variables

	ANOVA	SLDA	ANOVA	SLDA	ANOVA	SLDA	ANOVA	SLDA
3 genotypes	**Scenario 1**		**Scenario 2**		**Scenario 3**		**Scenario 4**	
Exactly right	5.5	13.5	1.5	64.0	91.5	85.5	89.5	49.0
False positives only	0.0	1.0	2.0	24.5	8.5	14.5	10.5	51.0
False negatives only	85.5	77.5	86.0	10.0	0.0	0.0	0.0	0.0
False positives and false negatives	9.0	8.0	10.5	1.5	0.0	0.0	0.0	0.0
Exact right + false positive	5.5	14.5	3.5	88.5	100.0	100.0	100.0	100.0
10 genotypes	**Scenario 5**		**Scenario 6**		**Scenario 7**		**Scenario 8**	
Exactly right	15.0	19.5	17.5	57.0	80.5	78.5	80.0	22.0
False positives only	1.0	2.5	4.5	36.0	14.0	19.5	14.5	78.0
False negatives only	72.5	64.5	71.0	7.0	5.0	1.5	4.5	0.0
False positives and false negatives	11.5	13.5	7.0	0.0	0.5	0.5	1.0	0.0
Exact right + false positive	16.0	22.0	22.0	93.0	94.5	98.0	94.5	100.0

5.2.2 Assessing Classification Methods

The next theme of interest in our simulation study is to assess the performance of the classification methods under various scenarios. Table 5-9 shows a summary of the classification results on the simulated linear data.

Regardless of the number of genotypes, all the algorithm perform significantly better than random chance. The rate of correct classification, however, declines as the number of groups increases. When genotypes are further apart (scenarios 3, 4, 7, and 8), the rate of correct classification is generally higher. When correlation among variables is high (scenarios 2, 4, 6, and 8), results are more impressive. The two linear classification methods, LDA and CDA, performed significantly better than did the nonlinear classification methods. GDA's poor performance, however, is rather surprising.

Table 5-9. Rate of Correct Classification for Linear Data

Correct Classification Rate Using Test Dataset								
	SVM		GDA		LDA		CDA	
	ANOVA	SLDA	ANOVA	SLDA	ANOVA	SLDA	ANOVA	SLDA
Scenario 1 (3 genotypes)	0.444	0.451	0.345	0.352	0.456	0.462	0.457	0.463
Scenario 2 (3 genotypes)	0.506	0.690	0.368	0.436	0.517	0.700	0.518	0.706
Scenario 3 (3 genotypes)	0.854	0.853	0.660	0.663	0.856	0.855	0.859	0.858
Scenario 4 (3 genotypes)	0.986	0.986	0.943	0.951	0.989	0.989	0.990	0.990
Scenario 5 (10 genotypes)	0.106	0.108	0.101	0.102	0.135	0.138	0.134	0.138
Scenario 6 (10 genotypes)	0.127	0.174	0.103	0.107	0.168	0.229	0.168	0.229
Scenario 7 (10 genotypes)	0.152	0.152	0.104	0.103	0.224	0.224	0.224	0.225
Scenario 8 (10 genotypes)	0.292	0.308	0.122	0.118	0.353	0.369	0.354	0.370

Correct Classification Rate Through Cross-Validation								
	SVM		GDA		LDA		CDA	
	ANOVA	SLDA	ANOVA	SLDA	ANOVA	SLDA	ANOVA	SLDA
Scenario 1 (3 genotypes)	0.4508	0.4571	0.3539	0.3549	0.5032	0.5146	0.5015	0.5126
Scenario 2 (3 genotypes)	0.4988	0.6703	0.3673	0.4154	0.5553	0.6937	0.5522	0.6936
Scenario 3 (3 genotypes)	0.8384	0.8386	0.5794	0.5868	0.8491	0.8514	0.8513	0.8518
Scenario 4 (3 genotypes)	0.9842	0.9858	0.9414	0.9481	0.9879	0.9896	0.9872	0.9898
Scenario 5 (10 genotypes)	0.1037	0.1045	0.1017	0.1044	0.1449	0.1472	0.1459	0.1476
Scenario 6 (10 genotypes)	0.1282	0.1774	0.1037	0.1047	0.1834	0.2405	0.1832	0.2389
Scenario 7 (10 genotypes)	0.1508	0.1458	0.1041	0.1034	0.2139	0.2149	0.2138	0.2150
Scenario 8 (10 genotypes)	0.2728	0.2798	0.1250	0.1226	0.3475	0.3652	0.3454	0.3640

Table 5-10 shows the rate of correct classification for simulated nonlinear datasets. As with the linear datasets, the number of groups and the distance between the groups influences the classification results. For example, the rates of correct classification under scenario 15 is much lower than in scenario14 due to the much larger number of groups to classify. Also, as the distance between groups centers increases among scenarios 9, 10, and 11, the classification results improves. Similarly, all methods did better in scenarios 14 than in scenario 13. Overall, SVM provided the best performance across all nonlinear types we have tried. GDA, on the other hand, in most cases still lagged behind all other methods in terms of the rate of correct classification. It came a very close second when the distances between the genotypes are large (Scenario 11). LAD turned out to be a very reasonable method to use even when the data is nonlinear. Its performance was second on all but two occasions. Although our experiment is very limited, combining results from the linear datasets, LDA should clearly be considered as a classification method in almost all situations.

Table 5-10. Rate of Correct Classification for Simulated Nonlinear Data

Correct Classification Rate Using Test Dataset	SVM	GDA	LDA	CDA
Scenario 9 (3 genotypes)	0.4738	0.3688	0.4261	0.3469
Scenario 10 (3 genotypes)	0.5892	0.5223	0.5511	0.3348
Scenario 11 (3 genotypes)	0.7069	0.6920	0.6685	0.3333
Scenario 12 (3 genotypes)	0.5976	0.4846	0.5960	0.3334
Scenario 13 (3 genotypes)	0.5464	0.3957	0.5183	0.5336
Scenario 14 (3 genotypes)	0.8541	0.7364	0.8039	0.7570
Scenario 15 (10 genotypes)	0.2219	0.1201	0.1860	0.2009

Correct Classification Rate Through Cross-Validation	SVM	GDA	LDA	CDA
Scenario 9 (3 genotypes)	0.4208	0.3767	0.4322	0.3446
Scenario 10 (3 genotypes)	0.5542	0.4467	0.5628	0.3338
Scenario 11 (3 genotypes)	0.7083	0.7075	0.6788	0.3333
Scenario 12 (3 genotypes)	0.5783	0.4317		
Scenario 13 (3 genotypes)	0.5100	0.3858		
Scenario 14 (3 genotypes)	0.8475	0.7542		

5.2.3 Validation Through Cross-Validation

Throughout the project, we have used cross-validation for the purpose of evaluating the performance of the classification methods when applied to real data. In order to assess this validation approach, we examine how it compared to the results obtained by using a test dataset. The generation of the test dataset is documented in Section 5-1. Table 5-11 summarizes the comparison of rate of correct classification obtained through the two methods. For most cases, the difference between rates produced by the two methods is very small. Statistical testing results show that none of the differences is statistically significant. Understanding the limitation of a single simulation scenario may have, we also calculated the average differences for all methods (see Table 5-12), and they are also statistically insignificant.

Table 5-11. Difference in Rates of Correct Classification between Test Dataset and Cross-Validation - Linear Data

	SVM		GDA		LDA		CDA	
	ANOVA	**SLDA**	**ANOVA**	**SLDA**	**ANOVA**	**SLDA**	**ANOVA**	**SLDA**
Scenario 1 (3 genotypes)	0.0065	0.0058	0.0084	0.0030	0.0476	0.0528	0.0449	0.0498
Scenario 2 (3 genotypes)	-0.0073	-0.0194	-0.0009	-0.0206	0.0387	-0.0061	0.0339	-0.0121
Scenario 3 (3 genotypes)	-0.0153	-0.0139	-0.0807	-0.0766	-0.0069	-0.0033	-0.0080	-0.0058
Scenario 4 (3 genotypes)	-0.0014	-0.0007	-0.0017	-0.0029	-0.0011	0.0004	-0.0027	-0.0002
Scenario 5 (10 genotypes)	-0.0025	-0.0032	0.0007	0.0029	0.0104	0.0093	0.0117	0.0100
Scenario 6 (10 genotypes)	0.0008	0.0034	0.0003	-0.0019	0.0156	0.0119	0.0154	0.0101
Scenario 7 (10 genotypes)	-0.0011	-0.0066	0.0003	-0.0001	-0.0097	-0.0093	-0.0103	-0.0099
Scenario 8 (10 genotypes)	-0.0196	-0.0286	0.0032	0.0048	-0.0054	-0.0039	-0.0083	-0.0061
Average	-0.0050	-0.0079	-0.0088	-0.0114	0.0112	0.0065	0.0096	0.0045

Table 5-12. Difference in Rates of Correct Classification between Test Dataset and Cross-Validation - Nonlinear Data

	SVM	**GDA**	**LDA**	**CDA**
Scenario 9 (3 genotypes)	0.0529	-0.0079	-0.0061	0.0024
Scenario 10 (3 genotypes)	0.0350	0.0757	-0.0117	0.0010
Scenario 11 (3 genotypes)	-0.0014	-0.0155	-0.0103	0.0000
Scenario 12 (3 genotypes)	0.0193	0.0530		
Scenario 13 (3 genotypes)	0.0364	0.0099		
Scenario 14 (3 genotypes)	0.0066	-0.0178		
Scenario 15 (3 genotypes)	0.2219	0.1201		
Average	0.0529	0.0311	-0.0094	0.0011

6.0 Application to Real Data

6.1 Comparison of Results

Both linear and nonlinear classification methods were applied to four datasets. Details of the datasets are described as follows:

- Draper Mouse Urine Dataset 5: This dataset consists of chromatograms of 67 urine samples from 20 mice of three strains. Among the 67 samples, 23 samples are from six AKR-H2k mice, 21 are from seven B6-H2b mice, and 23 are from seven B6-H2k mice. Samples were analyzed by Solid-Phase Microextraction (SPME) with GC/MS. Each sample has one replicate.

- Draper Human Plasma Dataset 7: This dataset consists of chromatograms of 170 plasma samples from 16 human donors of 13 unique HLA types. Each donor contributed up to three samples. Up to 13 replicate analyses were performed for each sample. Samples were analyzed by SPME with GC/MS.

- Monell Human Urine SPME Dataset: This dataset consists of chromatograms of 21 urine samples from seven human donors that can be grouped into two HLA-A supertypes. Because of the differences in other HLA genes (e.g., HLA-B, C, etc.), we chose to treat the seven donors as having seven unique HLA types. Each donor has three samples; two replicate analyses were performed for each sample. The samples were analyzed by SPME-GC/MS.

- KL Human Twin Sweat Dataset: This dataset consists of chromatograms of 62 sweat samples from 31 pairs of twins (22 identical twins and 9 fraternal twins). Each donor contributed one sample. One replicate analysis was performed for each sample. Samples were analyzed by SPME with GC/MS.

The steps used for these four datasets were similar: (1) chromatogram conversion and smoothing, if necessary; (2) component detection using AMDIS; (3) chromatogram alignment using multidimensional clustering method; (4) component quantification and normalization; (5) initial dimension reduction by elimination of impossible compounds; (6) further dimension reduction by statistical methods (ANOVA and SLDA); and (7) classification (LDA, CDA, GDA and SVM).

Tables 6-1 through 6-4 list the classification results for these four datasets.

Table 6-1. The rate of correct classification for Draper Mouse Urine Dataset 5

VOC Sets		Correct Classification Rates (%) by Cross Validation			
Set	**N**	**LDA**	**CDA**	**GDA**	**SVM**
Core – AI	151	79	31	66	78
Core – AI – E	91	64	46	67	79
Core – AI –SLDA	19	97	31	93	90
Core – AI – E – SLDA	14	94	31	90	94
Core – AI – ANOVA	61	69	33	82	79
Core – AI – E – ANOVA	37	72	42	81	81

Method artifacts (AI), environmental (E), and inconsistent elements identified using stepwise linear discriminate analysis (SLDA) or analysis of variance (ANOVA) were removed (subtracted) from the set of core chemicals prior to the classification using the indicated models. In the "n" column, the number of components remaining is indicated.

Table 6-2. The rate of correct classification for Draper Human Plasma Dataset 7

VOC Sets		Correct Classification Rates (%) by Cross Validation			
Set	**N**	**LDA**	**CDA**	**GDA**	**SVM**
Core – AI	46	42	31	39	44
Core – AI – E	10	36	29	36	42
Core – AI –SLDA	36	55	18	48	49
Core – AI – E – SLDA	8	38	29	31	42
Core – AI – ANOVA	40	43	31	40	49
Core – AI – E – ANOVA	9	36	29	34	43

Method artifacts (AI), environmental (E), and inconsistent elements identified using stepwise linear discriminate analysis (SLDA) or analysis of variance (ANOVA) were removed (subtracted) from the set of core chemicals prior to the classification using the indicated models. In the "n" column, the number of components remaining is indicated.

Table 6-3. The rate of correct classification for Monell Human Urine SPME Dataset

VOC Sets		Correct Classification Rates (%) by Cross Validation			
Set	**N**	**LDA**	**CDA**	**GDA**	**SVM**
Core – AI	415	90	48	90	93
Core – AI – E	147	79	29	83	83
Core – AI –SLDA	37	100	55	95	98
Core – AI – E – SLDA	41	93	29	74	93
Core – AI – ANOVA	309	93	50	90	95
Core – AI – E – ANOVA	104	88	14	81	86

Method artifacts (AI), environmental (E), and inconsistent elements identified using stepwise linear discriminate analysis (SLDA) or analysis of variance (ANOVA) were removed (subtracted) from the set of core chemicals prior to the classification using the indicated models. In the "n" column, the number of components remaining is indicated.

Table 6-4. The rate of correct classification for KL Human Twin Sweat Dataset

VOC Sets		Correct Classification Rates (%) by Cross Validation			
Set	**N**	**LDA**	**CDA**	**GDA**	**SVM**
Core – AI	100	14	11	5	7
Core – AI – E	37	7	2	9	7
Core – AI –SLDA	43	16	11	2	5
Core – AI – E – SLDA	6	34	9	16	25
Core – AI – ANOVA	44	30	18	25	36
Core – AI – E – ANOVA	14	11	7	11	11

Method artifacts (AI), environmental (E), and inconsistent elements identified using stepwise linear discriminate analysis (SLDA) or analysis of variance (ANOVA) were removed (subtracted) from the set of core chemicals prior to the classification using the indicated models. In the "n" column, the number of components remaining is indicated.

The results indicate that statistical dimension reduction improves classification. Among ANOVA and SLDA, SLDA fared better than ANOVA most of the time. This can be explained by the nature of SLDA, which considers the covariance structure of the variables.

Results also indicate that we successfully classified individual samples into the correct genotype 90% of the time when the number of genotypes was relatively small (less than 10). The method was less successful when applied to larger numbers of genotypes; however, the rates were still significantly higher than one would expect by chance alone. On the basis of these findings, we believe the success rates for larger numbers of genotypes would improve for larger sample sizes and greater precision among replicates.

Among the four statistical methods, it can be seen that between the linear methods of LDA and CDA, LDA was superior in classification of the samples. The nonlinear methods also performed well in all four datasets. Nonetheless, SVM consistently performed better than did GDA. In our assessments, SVM performed similarly to LDA in our cases due to the limited sample size and the high dimensionality of the data. In effect, the sample sizes were insufficient to adequately reveal nonlinearity in the data. Thus, our recommended approach is Step-wise Linear Discriminant Analysis to select important components, and then to use LDA when the sample size is small and SVM when the sample size is moderate or large for classification purposes.

6.2 Identified Compounds

Application of the statistical classification methods to the providers' data yielded a list of chemical species which exhibited discriminatory significance for a particular dataset. Table 6-5 illustrates the distribution of chemicals within compound class across datasets. For example, of the 104 compounds identified in human urine as having a contribution to the discriminating ability, 8% of those were acids.

Table 6-5. Discriminatory Chemical Identities, Expressed as Percent of Total

Compound Class	Human Urine	Mouse Urine	Human Plasma	Human Sweat
Acids	8	5	22	--
Aldehydes	5	8	33	7
Amines	6	16	--	--
Alcohols	26	14	11	29
Esters	1	11	--	--
Ethers	9	3	11	36
Hydrocarbons	7	8	11	21
Hydroxyketones	1	3	--	7
Ketones	24	16	11	--
S-containing	7	14	--	--
Other	8	3	--	--
Total Chemicals Identified	*104*	*37*	*9*	*14*

Overall, sample medium appears to have a profound affect on the types of compounds which demonstrate discriminatory influence. Alcohols and ketones are predominant chemical classes for human urine, accounting for 50% of the discriminatory compounds; mouse urine has a much flatter distribution, where only 30 % of the distribution is accounted for by alcohols and ketones, with approximately one-third the total number of discriminatory compounds as was found for human urine. Human plasma exhibited only nine compounds with discriminatory characteristics, of which three were aldehydes and two were acids. Unlike human urine, human blood did not have several compound classes—including amines, esters, hydroxyketones, and sulfur-containing compounds—as discriminating features. Somewhat similarly, human sweat had a relatively small number of total compounds (14) and exhibited the smallest number of chemical classes. Note that the results in Table 6-5 are derived from a relatively small number of samples. The human urine data are derived from seven donors from two HLA supertypes (which we treated as seven unique HLA types). The mouse urine data are derived from a total of 20 rats from three strains (and, thus, three MHC types). The human plasma data are derived from 16 donors with 13 unique HLA types. Finally, the sweat data are derived from 31 pairs of twins (22 identical, 9 fraternal sets). Refinements in the distributions of chemicals are expected as the numbers of samples increase and the identifications are confirmed based on the GC/MS analysis of authentic chemical standards.

The individual compound lists for each sample medium are presented in Tables 6-6 through 6-9. The specific chemical species displayed in these lists represent the summarized results of the actual NIST library-assigned names, across all samples in a dataset. Furthermore, only those compounds included in the variable selection are presented. In many instances, very specific structural isomers are named and no attempt, other than that in Table 6-5, has been made to substitute more generic compound names (e.g., “methyl-substituted heptanone” for “6-methyl-3-heptanone”). Unquestionably, the data processing scheme and the NIST identification contain sufficient uncertainty to warrant such a substitution, as the mass fragmentation patterns for structural isomers of a given compound are, in many cases, indistinguishable. Final confirmation of an identification in the unknown requires the co-elution of the compound with a known

chemical standard and that the unknown and the chemical standard each produce the same mass spectrum.

Table 6-6. Chemical Species Identified in the Human Urine Data

2(3H)-Furanone, 5-ethenyldihydro-5-methyl-
2-Octenoic acid, cis-
Benzene, ethoxy-
3-(Methylthio)-2-butanone
2-Pentanone, 4-hydroxy-
Naphthalene, 1,2-dihydro-1,1,6-trimethyl-
Furan, 2-ethyl-5-methyl-
3-Cyclohexene-1-methanol, .alpha.,.alpha.4-trimethyl-
2-Cyclohexen-1-ol, 2-methyl-5-(1-methylethenyl)-, cis-
2-Cyclopenten-1-one, 2-methyl-
Benzene, 1-ethenyl-4-methoxy-
o-Hydroxybiphenyl
5-Ethyl-2-furaldehyde
1-Cyclohexene-1-methanol, 4-(1-methylethenyl)-
Benzofuran, 4,7-dimethyl-
Indole
2-Buten-1-one, 1-(2,6,6-trimethyl-1,3-cyclohexadien-1-yl)-, (E)-
Phenol, 4-ethyl-2-methoxy-
3-Heptanone
2-Pentanone
2-Naphthalenemethanol, 2,3,4,4a,5,6,7,8-octahydro-.alpha.,.alpha.,4a,8-tetramethyl-
Oxirane, 2-(hexyn-1-yl)-3-methoxymethylene-
(+)-4-Carene
7-Octen-2-ol, 2,6-dimethyl-
cis-Z-.alpha.-Bisabolene epoxide
1-Adamantaneacetic acid
4-Oxepincarboxylic acid, 2,3,6,7-tetrahydro-, ethyl ester
trans-3-Caren-2-ol
4H-Imidazol-4-one, 2-amino-1,5-dihydro-
2,6,6-Trimethyl-2-cyclohexene-1,4-dione
Cyclobutanespiro-2'-bicyclo[1.1.0]butane-4'-spirocyclobutane
Nonanal
Hexanal
3-Heptanone, 6-methyl-
2H-Pyran-2-one, tetrahydro-6-methyl-
Octanal
Cyclohexanol, 1-methyl-4-(1-methylethenyl)-
3-Hexanone
Furan, 2,5-dimethyl-
Acetic acid
Benzenemethanol, 4-(1-methylethyl)-
Methanethiol
2,5-Furandione, 3-methyl-4-propyl-
Furan, 2,4-dimethyl-
Benzofuran, 2,3-dihydro-
Thiophene, 2-methoxy-

(continued)

Table 6-6. Continued

1,4-Cyclohexadiene-1-methanol, 4-(1-methylethyl)-
2-Heptanone
Ethanol, 2-phenoxy-
Propanoic acid, 2,2-dimethyl-
Dimethyl sulfone
2H-1-Benzopyran, 3,4,4a,5,6,8a-hexahydro-2,5,5,8a-tetramethyl-(2.alpha.,4a.alpha.,8
Propanoic acid, 2-methyl-
Sulfide, allyl methyl
Ethanone, 1-(1,4-dimethyl-3-cyclohexen-1-yl)-
Ethanone, 1-(4-methylphenyl)-
Benzeneacetaldehyde
Hexanoic acid
Tricyclo[2.2.1.0(2,6)]heptane-3-methanol, 2,3-dimethyl-
Cyclohexanone
3-Octanone
3,6-Heptanedione
Pyrazine, trimethyl-
3-Heptenoic acid
2(3H)-Furanone, dihydro-4,5-dimethyl-
Propanoic acid
Pyrazine, 2-ethenyl-6-methyl-
(+)-3-Carene, 10-(acetylmethyl)-
p-Mentha-1,5-dien-8-ol
.+/-.-4-Acetyl-1-methylcyclohexene
Ethanol, 2-(2-butoxyethoxy)-
Phenol
Formamide, N-phenyl-
S-Ethyl ethanethioate
1-Hexanol
2H-1-Benzopyran, 3,4,4a,5,6,8a-hexahydro-2,5,5,8a-tetramethyl-(2.alpha.,4a.alpha.,8
1,3-Benzodioxol-5-ol
2,2,6,6,-Tetramethylcyclohexanone
2H-1-Benzopyran, 3,5,6,8a-tetrahydro-2,5,5,8a-tetramethyl-, trans-
N-Butyl-tert-butylamine
2-Hexen-1-ol, (E)-
3-Penten-2-one, 4-methoxy-
2(3H)-Furanone, dihydro-3,5-dimethyl-
Benzene, 4-ethenyl-1,2-dimethoxy-
6-Hepten-3-one, 4-methyl-
3,4-Dimethylcyclopentanone
2-Pentanone, 3-ethyl-
Naphthalene, 1,2-dihydro-1,1,6-trimethyl-
1-Nonen-4-ol
1H-Indene, 2,3-dihydro-1,1,5,6-tetramethyl-
3-Cyclohexen-1-ol, 1-methyl-4-(1-methylethyl)-
2,6-Dimethyl-1,3,5,7-octatetraene, E,E-
2,6-Pyridinediamine
Epicedrol
trans,trans-3,5-Heptadien-2-one

(continued)

Table 6-6. Continued

1-Pentanol
2-Pentanone, 4-hydroxy-4-methyl-
1,3-Cyclohexadiene-1-methanol, 4-(1-methylethyl)-
2,6-Dimethyl-1,3,5,7-octatetraene, E,E-
Caprolactam
Propane, 1-(methylthio)-
4-Octanone
Tricyclo[4.4.0.0(2,7)]dec-8-ene-3-methanol, .alpha.,.alpha.,6,8-tetramethyl-, stere
Phenylethyl Alcohol

Table 6-7 . Chemical Species Identified in the Mouse Urine Data

Ethanone, 1-(4,5-dihydro-2-thiazolyl)-
Ethanone, 1-(1H-pyrrol-2-yl)-
Formamide, N-(2-methylphenyl)-
2-Amino-5-propyl-1,3,4-thiadiazole
Cyclohexanol, 2,6-dimethyl-
2-Penten-1-ol, acetate, (Z)-
o-Toluidine
1H-Pyrazolo[3,4-d]pyrimidine-4,6(5H,7H)-dione
Ethanol, 2-butoxy-
7-Exo-ethyl-5-methyl-6,8-dioxabicyclo[3.2.1]oct-3-ene
1-Dodecanol
2-Penten-1-ol, acetate, (Z)-
Benzyl methyl ketone
2-sec-Butylthiazole
1,3-Oxathiane, 2-isopropyl-2,6-dimethyl-
Propanoic acid, decyl ester
Furan, 2-ethyl-5-methyl-
Propanoic acid
Trimethylamine
7-Exo-ethyl-5-methyl-6,8-dioxabicyclo[3.2.1]oct-3-ene
6,8-Dioxabicyclo[3.2.1]octane, 7-ethyl-5-methyl-, (1R-exo)
Butanoic acid
3-Heptanone, 6-methyl-
n-Dodecyl acetate
Octanal
Ethanone, 1-(4,5-dihydro-2-thiazolyl)-
5,8-Decadien-2-one, 5,9-dimethyl-, (E)-
3-Cyclohexene-1-methanol, .alpha.,.alpha.4-trimethyl-
Thiourea
o-Xylene
Hexenal, 2-ethyl-
1-Hexadecanol
4-Octen-3-one, 6-ethyl-7-hydroxy-
Iminoformamide,N,N-dimethyl-N'-(3-methyl-2-oxotetrahydro-3
6-Hepten-3-one, 4-methyl-
Hexenal, 2-ethyl-
2-Piperidinone

Table 6-8. Chemical Species Identified in the Human Plasma Data

Octanoic Acid
5-Hepten-2-one, 6-methyl-
Hexanal
2,4-Dimethyl-1-heptene
Nonanal
Hexadecanal
3-Cyclohexen-1-ol, 4-methyl-1-(1-methylethyl)-
Propanoic acid, 2-methyl-, 1-(1,1-dimethylethyl)-2-methyl-
Furan, 2-pentyl-

Table 6-9. Chemical Species Identified in the Human Sweat Data

Decanal
2-Butanone, 3-hydroxy-
Ethanol, 2-(2-butoxyethoxy)-
2,6-Octadien-1-ol, 3,7-dimethyl-, (E)-
Acetic acid, phenylmethyl ester
Benzyl Benzoate
Cyclohexane, propyl-
n-Hexyl salicylate
Benzeneethanol, .alpha.,.alpha.-dimethyl-, acetate
1,6-Octadien-3-ol, 3,7-dimethyl-
4-tert-Butylcyclohexyl acetate
Dodecane
Ethanol, 2-butoxy-
Decane

7.0 References

Baudat, G., and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385-2404, 2000.

Franc, V., and V. Hlavac. "Statistical Pattern Recognition Toolbox for Matlab: User's Guide." Czech Technical University.

Mika, S., G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher Discriminant Analysis with Kernels. In Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41-48. IEEE, 1999.

Müller, K. R., S. Mika, G. Rätsch, and K. Tsuda. An Introduction to Kernel-based Learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181-201, 2001.

Shen, H.; Grung, B.; Kvalheim, O.; and Eide, I. *Analytica Chimica Acta* 446 (2001) 313-328.

Shawe-Taylor, John, and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Windig, W.; Phalp, J.M.; and Payne, A.W. *Analytical Chemistry* 68 (1996) 3602-3606.

Reports/Presentations Prepared During the Project

1. Monthly reports from July 2003 through February 2005.
2. Quarterly Report Presentations:
    a. October 2003
    b. February 2004
    c. June 2004
    d. October 2004
3. Semi-Annual Reports
    a. January 2004
    b. August 2004
4. Report on Statistical Analysis Workshop held at RTI in May 2004
5. Annual Report March 2004
6. USD Milestone Presentation October 2004
7. USD PI Meeting presentations
    a. August 2003
    b. April 2004
    c. August 2004
    d. January 2005

There was also one poster presented at a meeting:

Raymer, J. H ., J. Deese-Spruill, T. Marrero, M. Rice. *Collection and Analysis of Volatile Organic Compounds (VOCs) from Human Sweat.* Presented at the Annual Meeting of the ISEA, Philadelphia, October 2004.

Appendix A

Instructions for Operation of the Package

Operation Instructions

This document contains instructions for processing and analyzing each experiment's data. These instructions assume that the user:

- is familiar with our data processing and analysis procedures as described in the Final Technical Report,
- has access to and is familiar with the software packages and programming languages used in these procedures, including:
    - Microsoft Excel XP;
    - SAS version 8.02;
    - Matlab release 14 (some programs were executed in release 11, but should be forward compatible to release 14)
    - NIST/EPA/NIH Mass Spectral Library (NIST 02),
    - NIST Mass Spectral Search Program version 2.0a,
    - AMDIS version 2.6, and
    - GC/MS instrument software; and
- has access to the files described in this document.

All programs have been run under the Microsoft Windows environment—either Windows 2000 or Windows XP Professional; no other operating systems have been tested.

Table A-1. Processing and Analysis Steps for Each Experiment

Step	Processing/Analysis Step Description	Relevant Files or Software	Parameters or Settings
1	Convert the raw data files to CDF format. **[2.b.i.]**	instrument software	software dependent
2	Complete the experiment map. The experiment map assigns an analysis file name to every raw data file in the experiment. The analysis file name follows a common naming convention across all experiments and incorporates elements of the experimental design, such as genetic type, donor ID, and sample number. The map also lists the directory locations of all raw data files and CDF data files.	dcc\Konrad_Lorenz\Analysis\KL Experiment Map.xls	n/a
3	Smooth the data if necessary. **[2.b.iii.]**		
	a Convert all the CDF files to Matlab.	dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\convert_matlab*.m	none
	b Smooth the data and output revised CDF data files.	dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\smooth_all*.m	Set the parameters for the Savitzky-Golay smoothing algorithm in the call to the *sgolayfilt* function.
4	Run AMDIS on all the CDF data files. **[2.b.ii.]**	AMDIS version 2.6	We ran our analyses with the following deconvolution settings: Resolution = Low, Sensitivity = Very Low, Shape requirements = Medium. We also set the Analysis Type to Simple. All other settings remained at their defaults.
5	Create a DOS batch file that will copy all the AMDIS-generated files into the Analysis folder and rename them using the appropriate analysis filename.	01_create_batch_file.sas	None
6	Run the batch file created in the previous step.	02_copy_amdis_output.bat	None
7	Parse all the AMDIS output and create a single SAS dataset containing the results.	03_process_components.sas	Several parameters can be set when parsing the AMDIS output. See the file dcc\production\processing\parse_amdis_component.sas for a full list. The most important parameters to set are the minimum and maximum m/z values in the file, which are set in the call to the %parse_amdis_components macro. We recommend leaving the other parameters set the way they are in the 03_process_components.sas file.
8	Locate the internal standard(s) in all the files. **[2.b.iv.1.]**	04_find_internal_standards.sas	For each standard, enter the expected retention time, the retention time search window, and the expected mass spectrum.

Step	Processing/Analysis Step Description	Relevant Files or Software	Parameters or Settings
9	While searching for internal standards, if any files are identified as being "bad" files for some reason, remove them from the analysis. **[2.b.iv.1.]**	05_remove_bad_files.sas	Enter the list of filenames that should be removed from analysis.
10	Locate a few landmarks, which are components detected in most of the files but are not internal standards. **[2.b.iv.1.]**	06_find_landmarks.sas	For each landmark, enter the expected retention time, the retention time search window, and the expected mass spectrum.
11	Perform a coarse alignment by stretching and compressing each of the files so that the internal standards and landmarks have exactly the same retention time across all files. **[2.b.iv.2.]**	07_coarse_alignment.sas	None
12	Use multidimensional clustering to match components across files, creating clusters of like components. **[2.b.iv.3.]**	08_cluster_components.sas	Set the retention time weight, the cluster radius, and the maximum number of clusters in the call to the %multidimensional_clustering macro.
13	Normalize each component's area by the area of the internal standard. **[2.b.viii.]**	09_quantify_components.sas	None
14	Find the core clusters; use a variety of consistency criteria, if applicable. **[2.b.v.]**	10_find_core_components.sas	Define the genetic types, donors, samples, and replicates at the top of the program. Define the range of consistency criteria in the call to the %loop_core macro.
15	Create text files containing the spectra of all components in the core clusters. These files will be used as input into the NIST library search.	11_find_core_spectra.sas	None
16	Run the spectra files generated in the previous step through the NIST library search. Output the search results to text files. **[2.b.vi.]**	NIST Mass Spectral Search Program version 2.0a	Set the system printer to "Generic/Text Only" and orientation to "Landscape." Select Library Search Options under Tools and select the Automation tab. Set Number of Hits to Print to "2." In the same dialog, select the Search tab and select the Automation and Auto Report checkboxes. In the main program view, select the "Lib. Search" tab and open the appropriate *.msp file. Select "Import All" in the dialog box and "Overwrite the Spec List contents." One output file will be created for each spectral search.
17	Parse all the NIST search output and create a single SAS dataset containing the results.	11a_nistsearch2.sas	Set the number of input files at the top of the program.

Step	Processing/Analysis Step Description	Relevant Files or Software	Parameters or Settings
18	For each cluster, produce a frequency distribution showing how often each NIST compound was a search result. **[2.b.vii.]**	11b_freq_nisttest.sas	None
19	Decide which clusters should be flagged as impossible or environmental. **[2.b.vii., 3.a.]**	dcc\Michael\Code\nistfreq.sas; dcc\Michael\Code\environ.wk3	Define the SAS dataset to be imported at the top of the nistfreq.sas program.
20	Create a dataset indicating which clusters are flagged as impossible, environmental, and heterogeneous. **[2.b.vii., 3.a.]**	11c_filtered_clusters.sas	None
21	Perform variable selection using ANOVA. **[3.b.i.]**	12_reduction_anova.sas	None
22	Perform variable selection using SLDA. **[3.b.ii.]**	15_reduction_slda.sas	Set the number of core clusters at the top of the program.
23	Find correct classification rates with cross-validation using LDA. **[4.a.i.]**	15_lda_cross.sas	Set the number of core clusters at the top of the program.
24	Find correct classification rates with cross-validation using CDA. **[4.a.ii.]**	15_cda_cross.sas	Set the number of core clusters at the top of the program.
25	Create training and testing files for use with SVM and GDA, which are implemented in Matlab.	dcc\Development\Analysis\Test_data\KL01_data\CV_log_transform\CV_KL01data_log_transform.sas	none
26	Find correct classification rates with cross-validation using SVM. **[4.b.i.]**	dcc\Development\Analysis\Test_data\KL01_data\CV_log_transform\SVM\SVM_KL01_CV_log_transform.m	Set the options for the SVM algorithm in the lines immediately before the calls to the *oaasvm* function.
27	Find correct classification rates with cross-validation using GDA. **[4.b.ii.]**	dcc\Development\Analysis\Test_data\KL01_data\CV_log_transform\GDA\GDA_KL01_CV_log_transform.m	Set the options for the GDA algorithm in the lines immediately before the calls to the *gda* function.

Notes:

- Section references from the Final Technical Report are listed in brackets after applicable step descriptions.
- The programs listed here apply to the KL Twin Sweat data, which we have designated as KL Experiment #1. This entire set of programs (with the exception of the smoothing step) has also been run for Monell Experiment #4, Draper Experiment #5, and Draper Experiment #7. Programs written for those experiments are included in the software transfer but are not described here because they are very similar.
- Files are located in dcc\Konrad_Lorenz\Analysis\Experiment01 unless the full path name is provided.
- Numeric prefixes in file names and the file names themselves may be slightly different for other experiments.
- Input and output files are listed in program headers. In addition, most of the programs require entering the input and output files near the top or bottom of the program. This is assumed in the Parameters or Settings column and is not be repeated for every row of the table.

Appendix B

List of Software

Table B-1. File Names, Locations, and Descriptions

File Name and Location	File Description
Contents.doc	Lists all the files included in the delivery.
Operation Instructions.doc	Provides instructions for using these files to process and analyze data.
dcc\libnames.inc	Defines folder locations and SAS libnames for processing and analysis tasks.
dcc\Development\Analysis\Test_data\D05_data\CV_log_transform\CV_D05data_log_transform.sas	Creates training and testing files for use with SVM and GDA.
dcc\Development\Analysis\Test_data\D05_data\CV_log_transform\GDA\GDA_D05_CV_log_transform.m	Finds correct classification rates with cross-validation using GDA.
dcc\Development\Analysis\Test_data\D05_data\CV_log_transform\SVM\SVM_D05_CV_log_transform.m	Finds correct classification rates with cross-validation using SVM.
dcc\Development\Analysis\Test_data\D07_data\CV_log_transform\CV_D07data_log_transform.sas	Creates training and testing files for use with SVM and GDA.
dcc\Development\Analysis\Test_data\D07_data\CV_log_transform\GDA\GDA_D07_CV_log_transform.m	Finds correct classification rates with cross-validation using GDA.
dcc\Development\Analysis\Test_data\D07_data\CV_log_transform\SVM\SVM_D07_CV_log_transform.m	Finds correct classification rates with cross-validation using SVM.
dcc\Development\Analysis\Test_data\KL01_data\CV_log_transform\CV_KL01data_log_transform.sas	Creates training and testing files for use with SVM and GDA.
dcc\Development\Analysis\Test_data\KL01_data\CV_log_transform\GDA\GDA_KL01_CV_log_transform.m	Finds correct classification rates with cross-validation using GDA.
dcc\Development\Analysis\Test_data\KL01_data\CV_log_transform\SVM\SVM_KL01_CV_log_transform.m	Finds correct classification rates with cross-validation using SVM.
dcc\Development\Analysis\Test_data\M04_data\CV_log_transform\CV_M04data_log_transform.sas	Creates training and testing files for use with SVM and GDA.
dcc\Development\Analysis\Test_data\M04_data\CV_log_transform\GDA\GDA_M04_CV_log_transform.m	Finds correct classification rates with cross-validation using GDA.
dcc\Development\Analysis\Test_data\M04_data\CV_log_transform\SVM\SVM_M04_CV_log_transform.m	Finds correct classification rates with cross-validation using SVM.
dcc\Draper\Analysis\Draper Experiment Map.xls	Provides an experiment map for Draper.

File Name and Location	File Description
dcc\Draper\Analysis\Experiment05\01_create_batch_file.sas	Creates a DOS batch file that will copy all the AMDIS-generated files into the Analysis folder and rename them using the appropriate analysis file name.
dcc\Draper\Analysis\Experiment05\02_copy_amdis_output.bat	Provides the batch file created in the previous step.
dcc\Draper\Analysis\Experiment05\03_process_components.sas	Parses all the AMDIS output and create a single SAS dataset containing the results.
dcc\Draper\Analysis\Experiment05\04_find_internal_standards.sas	Locates the internal standard(s) in all the files.
dcc\Draper\Analysis\Experiment05\05_remove_bad_files.sas	Removes any "bad" files from the analysis.
dcc\Draper\Analysis\Experiment05\06_find_landmarks.sas	Locates a few landmarks, which are components detected in most of the files but are not internal standards.
dcc\Draper\Analysis\Experiment05\07_coarse_alignment.sas	Performs a coarse alignment by stretching and compressing each of the files so that the internal standards and landmarks have exactly the same retention time across all files.
dcc\Draper\Analysis\Experiment05\08_cluster_components.sas	Uses multidimensional clustering to match components across files, creating clusters of like components.
dcc\Draper\Analysis\Experiment05\09_quantify_components.sas	Normalizes each component's area by the area of the internal standard.
dcc\Draper\Analysis\Experiment05\10_find_core_components.sas	Finds the core clusters; uses a variety of consistency criteria, if applicable.
dcc\Draper\Analysis\Experiment05\11_find_core_spectra.sas	Creates text files containing the spectra of all components in the core clusters. These files will be used as input into the NIST library search.
dcc\Draper\Analysis\Experiment05\12_nist_search.sas	Parses all the NIST search output and creates a single SAS dataset containing the results.
dcc\Draper\Analysis\Experiment05\13_freq_nist.sas	For each cluster, produces a frequency distribution showing how often each NIST compound was a search result.
dcc\Draper\Analysis\Experiment05\14_filtered_clusters.sas	Creates a dataset indicating which clusters are flagged as impossible, environmental, and heterogeneous.
dcc\Draper\Analysis\Experiment05\15_reduction_anova.sas	Performs variable selection using ANOVA.
dcc\Draper\Analysis\Experiment05\16_cda_cross.sas	Finds correct classification rates with cross-validation using CDA.
dcc\Draper\Analysis\Experiment05\16_lda_cross.sas	Finds correct classification rates with cross-validation using LDA.
dcc\Draper\Analysis\Experiment05\16_reduction_SLDA.sas	Performs variable selection using SLDA.
dcc\Draper\Analysis\Experiment07\01_create_batch_file.sas	Creates a DOS batch file that will copy all the AMDIS-generated files into the Analysis folder and rename them using the appropriate analysis file name.
dcc\Draper\Analysis\Experiment07\02_copy_amdis_output.bat	Provides the batch file created in the previous step.
dcc\Draper\Analysis\Experiment07\03_process_components.sas	Parses all the AMDIS output and create a single SAS dataset containing the results.
dcc\Draper\Analysis\Experiment07\04_find_internal_standards.sas	Locates the internal standard(s) in all the files.
dcc\Draper\Analysis\Experiment07\05_remove_bad_files.sas	Removes any "bad" files from the analysis.

File Name and Location	File Description
dcc\Draper\Analysis\Experiment07\06_find_landmarks.sas	Locates a few landmarks, which are components detected in most of the files but are not internal standards.
dcc\Draper\Analysis\Experiment07\07_coarse_alignment.sas	Performs a coarse alignment by stretching and compressing each of the files so that the internal standards and landmarks have exactly the same retention time across all files.
dcc\Draper\Analysis\Experiment07\08_cluster_components.sas	Uses multidimensional clustering to match components across files, creating clusters of like components.
dcc\Draper\Analysis\Experiment07\09_quantify_components.sas	Normalizes each component's area by the area of the internal standard.
dcc\Draper\Analysis\Experiment07\10_find_core_components.sas	Finds the core clusters. Uses a variety of consistency criteria, if applicable.
dcc\Draper\Analysis\Experiment07\11_find_core_spectra.sas	Creates text files containing the spectra of all components in the core clusters. These files will be used as input into the NIST library search.
dcc\Draper\Analysis\Experiment07\11a_nistsearch2.sas	Parses all the NIST search output and creates a single SAS dataset containing the results.
dcc\Draper\Analysis\Experiment07\11b_freq_nisttest.sas	For each cluster, produces a frequency distribution showing how often each NIST compound was a search result.
dcc\Draper\Analysis\Experiment07\11c_filtered_clusters.sas	Creates a dataset indicating which clusters are flagged as impossible, environmental, and heterogeneous.
dcc\Draper\Analysis\Experiment07\13_cda_cross.sas	Finds correct classification rates with cross-validation using CDA.
dcc\Draper\Analysis\Experiment07\13_lda_cross.sas	Finds correct classification rates with cross-validation using LDA.
dcc\Draper\Analysis\Experiment07\13_reduction_slda.sas	Performs variable selection using SLDA.
dcc\Draper\Analysis\Experiment07\14_reduction_anova.sas	Performs variable selection using ANOVA.
dcc\Konrad_Lorenz\Analysis\KL Experiment Map.xls	Provides an experiment map for Konrad Lorenz.
dcc\Konrad_Lorenz\Analysis\Experiment01\01_create_batch_file.sas	Creates a DOS batch file that will copy all the AMDIS-generated files into the Analysis folder and rename them using the appropriate analysis file name.
dcc\Konrad_Lorenz\Analysis\Experiment01\02_copy_amdis_output.bat	Provides the batch file created in the previous step.
dcc\Konrad_Lorenz\Analysis\Experiment01\03_process_components.sas	Parses all the AMDIS output and create a single SAS dataset containing the results.
dcc\Konrad_Lorenz\Analysis\Experiment01\04_find_internal_standards.sas	Locates the internal standard(s) in all the files.
dcc\Konrad_Lorenz\Analysis\Experiment01\05_remove_bad_files.sas	Removes any "bad" files from the analysis.
dcc\Konrad_Lorenz\Analysis\Experiment01\06_find_landmarks.sas	Locates a few landmarks, which are components detected in most of the files but are not internal standards.
dcc\Konrad_Lorenz\Analysis\Experiment01\07_coarse_alignment.sas	Performs a coarse alignment by stretching and compressing each of the files so that the internal standards and landmarks have exactly the same retention time across all files.

File Name and Location	File Description
dcc\Konrad_Lorenz\Analysis\Experiment01\08_cluster_components.sas	Uses multidimensional clustering to match components across files, creating clusters of like components.
dcc\Konrad_Lorenz\Analysis\Experiment01\09_quantify_components.sas	Normalizes each component's area by the area of the internal standard.
dcc\Konrad_Lorenz\Analysis\Experiment01\10_find_core_components.sas	Finds the core clusters; uses a variety of consistency criteria, if applicable.
dcc\Konrad_Lorenz\Analysis\Experiment01\11_find_core_spectra.sas	Creates text files containing the spectra of all components in the core clusters. These files will be used as input into the NIST library search.
dcc\Konrad_Lorenz\Analysis\Experiment01\11a_nistsearch2.sas	Parses all the NIST search output and creates a single SAS dataset containing the results.
dcc\Konrad_Lorenz\Analysis\Experiment01\11b_freq_nisttest.sas	For each cluster, produces a frequency distribution showing how often each NIST compound was a search result.
dcc\Konrad_Lorenz\Analysis\Experiment01\11c_filtered_clusters.sas	Creates a dataset indicating which clusters are flagged as impossible, environmental, and heterogeneous.
dcc\Konrad_Lorenz\Analysis\Experiment01\12_reduction_anova.sas	Performs variable selection using ANOVA.
dcc\Konrad_Lorenz\Analysis\Experiment01\15_cda_cross.sas	Finds correct classification rates with cross-validation using CDA.
dcc\Konrad_Lorenz\Analysis\Experiment01\15_lda_cross.sas	Finds correct classification rates with cross-validation using LDA.
dcc\Konrad_Lorenz\Analysis\Experiment01\15_reduction_slda.sas	Performs variable selection using SLDA.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\convert_matlab1.m	Converts CDF files to Matlab.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\convert_matlab2.m	Converts CDF files to Matlab.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\convert_matlab3.m	Converts CDF files to Matlab.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\convert_matlab4.m	Converts CDF files to Matlab.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\convert_matlab5.m	Converts CDF files to Matlab.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\smooth_all.m	Smooths the data and outputs revised CDF data files.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\smooth_all2.m	Smooths the data and outputs revised CDF data files.
dcc\Konrad_Lorenz\Analysis\Experiment01\Smoothing\smooth_all3.m	Smooths the data and outputs revised CDF data files.
dcc\Michael\Code\environ.wk3	Provides a database of compound names that are considered environmental.
dcc\Michael\Code\NistFreq.sas	Decides which clusters should be flagged as impossible or environmental.
dcc\Monell\Analysis\Monell Experiment Map.xls	Provides an experiment map for Monell.
dcc\Monell\Analysis\Experiment04\01_create_batch_file.sas	Creates a DOS batch file that will copy all the AMDIS-generated files into the Analysis folder and rename them using the appropriate analysis file name.
dcc\Monell\Analysis\Experiment04\02_copy_amdis_output.bat	Provides the batch file created in the previous step.
dcc\Monell\Analysis\Experiment04\03_process_components.sas	Parses all the AMDIS output and create a single SAS dataset containing the results.
dcc\Monell\Analysis\Experiment04\04_find_internal_standards.sas	Locates the internal standard(s) in all the files.

File Name and Location	File Description
dcc\Monell\Analysis\Experiment04\05_remove_bad_files.sas	Removes any "bad" files from the analysis.
dcc\Monell\Analysis\Experiment04\06_find_landmarks.sas	Locates a few landmarks, which are components detected in most of the files but are not internal standards.
dcc\Monell\Analysis\Experiment04\07_coarse_alignment.sas	Performs a coarse alignment by stretching and compressing each of the files so that the internal standards and landmarks have exactly the same retention time across all files.
dcc\Monell\Analysis\Experiment04\08_cluster_components.sas	Uses multidimensional clustering to match components across files, creating clusters of like components.
dcc\Monell\Analysis\Experiment04\09_quantify_components.sas	Normalizes each component's area by the area of the internal standard.
dcc\Monell\Analysis\Experiment04\10_find_core_components.sas	Finds the core clusters; uses a variety of consistency criteria, if applicable.
dcc\Monell\Analysis\Experiment04\11_find_core_spectra.sas	Creates text files containing the spectra of all components in the core clusters. These files will be used as input into the NIST library search.
dcc\Monell\Analysis\Experiment04\12_nist_search.sas	Parses all the NIST search output and creates a single SAS dataset containing the results.
dcc\Monell\Analysis\Experiment04\13_freq_nist.sas	For each cluster, produces a frequency distribution showing how often each NIST compound was a search result.
dcc\Monell\Analysis\Experiment04\14_filtered_clusters.sas	Creates a dataset indicating which clusters are flagged as impossible, environmental, and heterogeneous.
dcc\Monell\Analysis\Experiment04\15_reduction_anova.sas	Performs variable selection using ANOVA.
dcc\Monell\Analysis\Experiment04\16_cda_cross.sas	Finds correct classification rates with cross-validation using CDA.
dcc\Monell\Analysis\Experiment04\16_lda_cross.sas	Finds correct classification rates with cross-validation using LDA.
dcc\Monell\Analysis\Experiment04\16_reduction_slda.sas	Performs variable selection using SLDA.
dcc\Production\macro_dde_excel.sas	Contains macros for working with Excel workbooks.
dcc\Production\macro_stringops.sas	Contains macros for dealing with lists of variable names.
dcc\Production\Processing\check_cluster.sas	Contains a diagnostic macro for checking the results of the multidimensional clustering.
dcc\Production\Processing\check_cluster_replicates.sas	Contains a diagnostic macro for checking the results of the multidimensional clustering.
dcc\Production\Processing\coarse_alignment.sas	Contains a macro for performing the coarse alignment procedure.
dcc\Production\Processing\convert_cdf.m	Contains a function for converting a CDF file to a Matlab dataset and a text file.
dcc\Production\Processing\convert_mat_to_txt.m	Contains a function for converting an intensity matrix from a Matlab dataset to a text file.
dcc\Production\Processing\COPYING	Contains the GNU General Public License, the license under which the sgolayfilt.m file was released.

File Name and Location	File Description
dcc\Production\Processing\copy_amdis_output.sas	Contains a macro for creating the DOS batch file that copies all the AMDIS-generated files into the Analysis folder and renames them using the appropriate analysis file name.
dcc\Production\Processing\find_target.sas	Contains a macro for searching AMDIS results for a particular compound.
dcc\Production\Processing\multidimensional_clustering.sas	Contains a macro for performing the multidimensional clustering procedure.
dcc\Production\Processing\parse_amdis_components.sas	Contains a macro for parsing .elu files output by AMDIS.
dcc\Production\Processing\sgolayfilt.m	Contains a function for performing Savitzky-Golay smoothing.
dcc\Production\Processing\write_revised_cdf.m	Contains a function for revising the intensity matrix of a CDF file.